

Azure Identity and Access Management tasks

author:Dibrova Volodymyr

Practical Task 1: Introduction to Microsoft Entra ID

Create a basic Microsoft Entra ID setup for an organization to manage identity and access.

Requirements:

1. Create a new Microsoft Entra ID tenant.
- 2.
3. Add at least two users to the directory.
4. Create two groups named **Developers** and **Admins**.
5. Assign the users to appropriate groups.
6. Assign the **Global Reader** role to the **Admins** group.
7. Assign the **Application Developer** role to the **Developers** group.

Actions Taken:

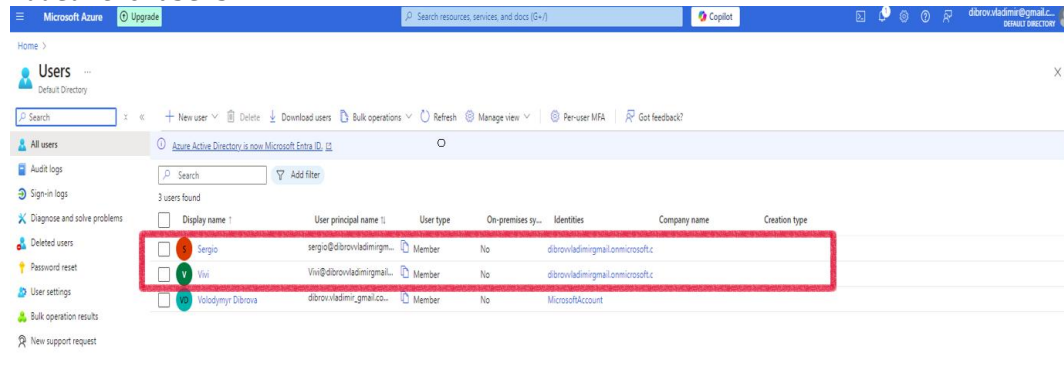
- Used Default Microsoft Entra ID instead of creating a new tenant due to issues with phone number registration.

The screenshot displays the Microsoft Azure portal interface for managing tenants. The main section, 'Manage tenants', shows a table of current tenants. The 'Default Directory (Default)' tenant is selected and highlighted with a red box. The 'Tenant details' sidebar on the right provides information about the selected tenant, including its name, ID, and a table of objects. The 'Objects' table is also highlighted with a red box.

Organization name	Domain name	Tenant type	Org ID
Default Directory (Default)	dibrovvladimir@gmail.onmicrosoft.com	Microsoft Entra ID	cd0728b4-be...
My test organization	mytestorg24.onmicrosoft.com	Azure AD B2C	a7f3a...

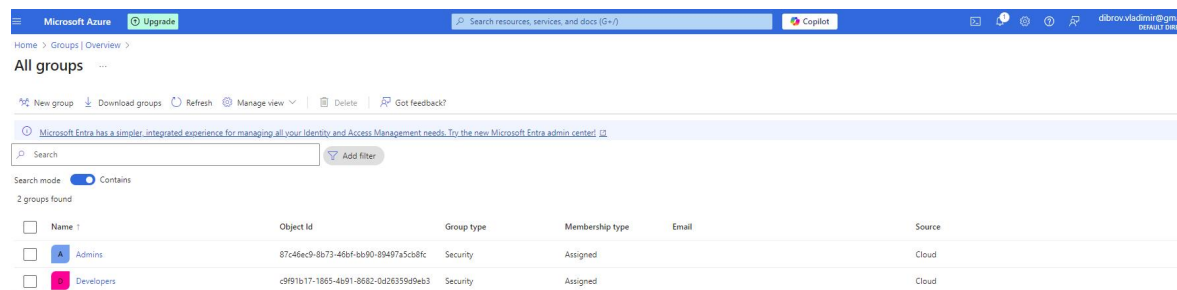
Object	Count
Users	4
Groups	2
Applications	5
Devices	0

- Added two users



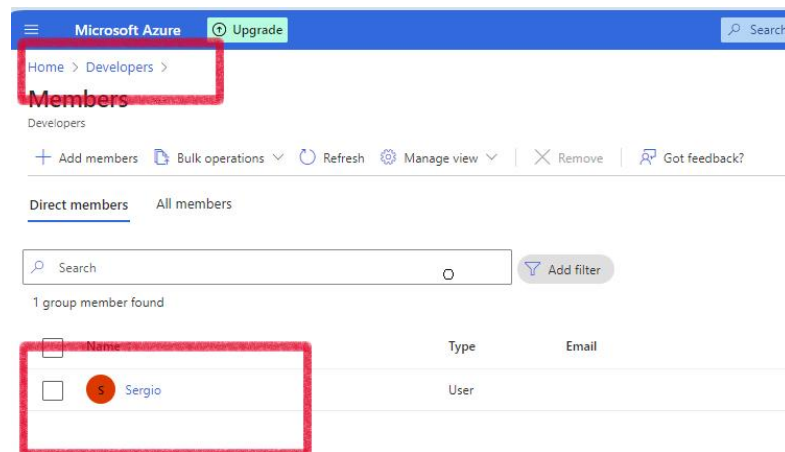
- Created two groups:

1. Developers
2. Admins

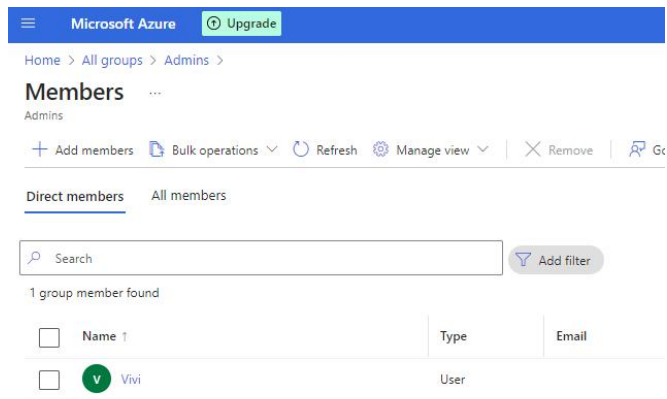


- Assigned users to the appropriate groups:

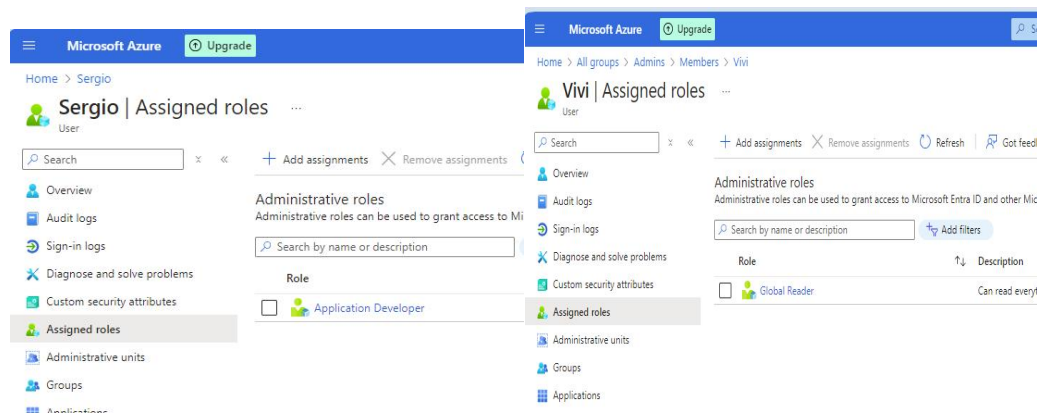
1. Developers: Sergio



2. Admins: Vivi



- Assigned roles:
- The first version



- The second version:
 - Reader role was assigned to the Admins group instead of the Global Reader role
 - App Service Environment Contributor role was assigned to the Developers group instead of the Application Developer role.

ps > MyLesson1

MyLesson1 | Access control (IAM)

Search

9 items (3 Users, 4 Groups, 1 Unknown, 1 Managed Identities)

Name	Type	Role	Scope	Condition
Owner (4)				
MyIdentity	User-assigned Managed Identity	Owner	Subscription (Inherited)	None
Volodymyr Dibrov	User	Owner	Subscription (Inherited)	None
Volodymyr Dibrov	User	Owner	Subscription (Inherited)	None
Identity not found	Unknown	Owner	Subscription (Inherited)	None
App Service Environment Contributor (1)				
Developers	Group	App Service Environment Contributor	This resource	None
Key Vault Administrator (1)				
Volodymyr Dibrov	User	Key Vault Administrator	Subscription (Inherited)	None
Key Vault Contributor (1)				
Admins	Group	Key Vault Contributor	This resource	None
Reader (1)				
Admins	Group	Reader	Subscription (Inherited)	None
Resource Viewer (1)				
Developers	Group	Resource Viewer	This resource	None

Verified test(Try to create a new resource by user Vivi)

Errors

Summary Raw Error

ERROR DETAILS

The client 'Vivi@dibrovvladimir@gmail.onmicrosoft.com' with object id '4818e660-bfd8-4344-bbb0-aaf2f813f078' does not have authorization to perform action 'Microsoft.Resources/tags/write' over scope '/subscriptions/507cbe71-1145-4a77-bdb8-d6fa9921aed5/resourceGroups/MyLesson1/providers/Microsoft.KeyVault/vaults/my-VN/providers/Microsoft.Resources/tags/default' or the scope is invalid. If access was recently granted, please refresh your credentials.

(Code: AuthorizationFailed)

WAS THIS HELPFUL?

Explain with Copilot

Troubleshooting Options

New Support Request

Implementation Highlights:

- Using Default Entra ID allowed bypassing registration restrictions and successfully completing the task.

Practical Task 2: Enabling Single Sign-On (SSO) and Multi-Factor Authentication (MFA)

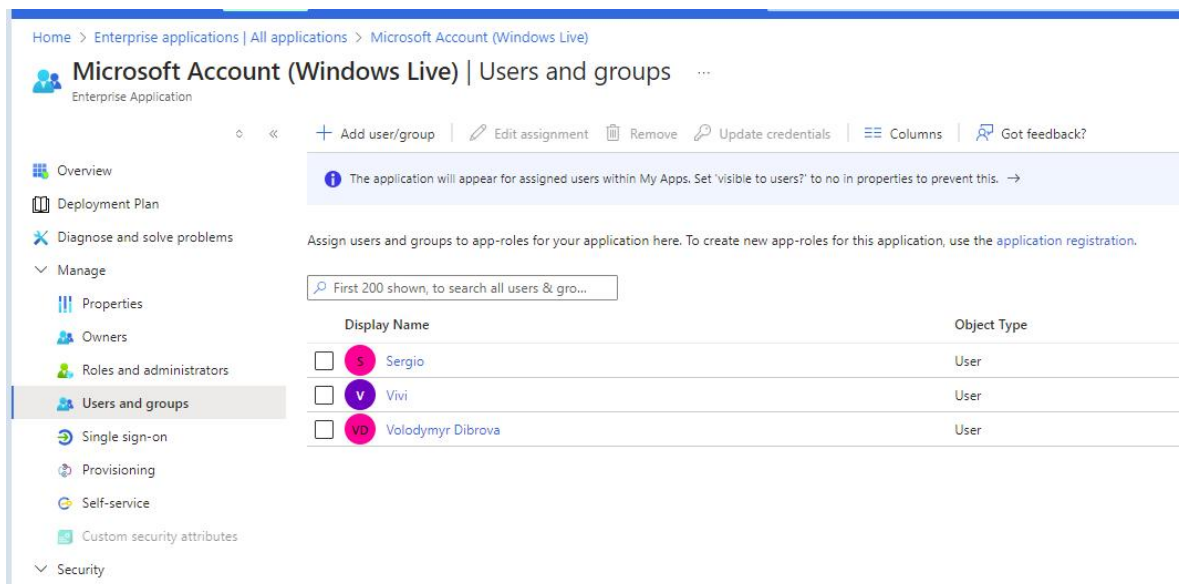
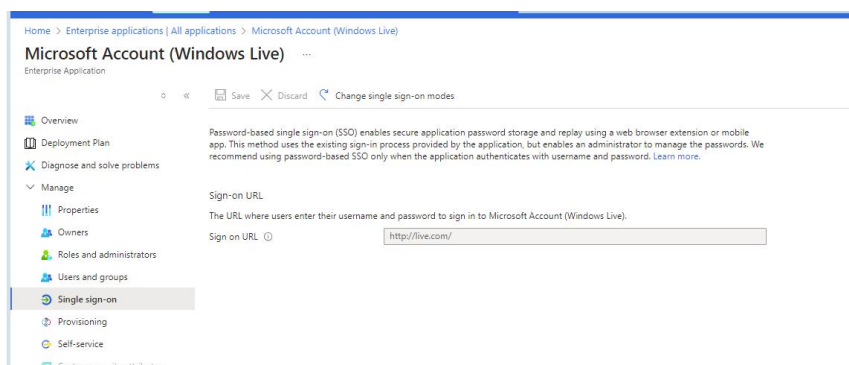
Configure Single Sign-On (SSO) and Multi-Factor Authentication (MFA) for users in a Microsoft Entra ID directory to enhance identity and access security.

Requirements:

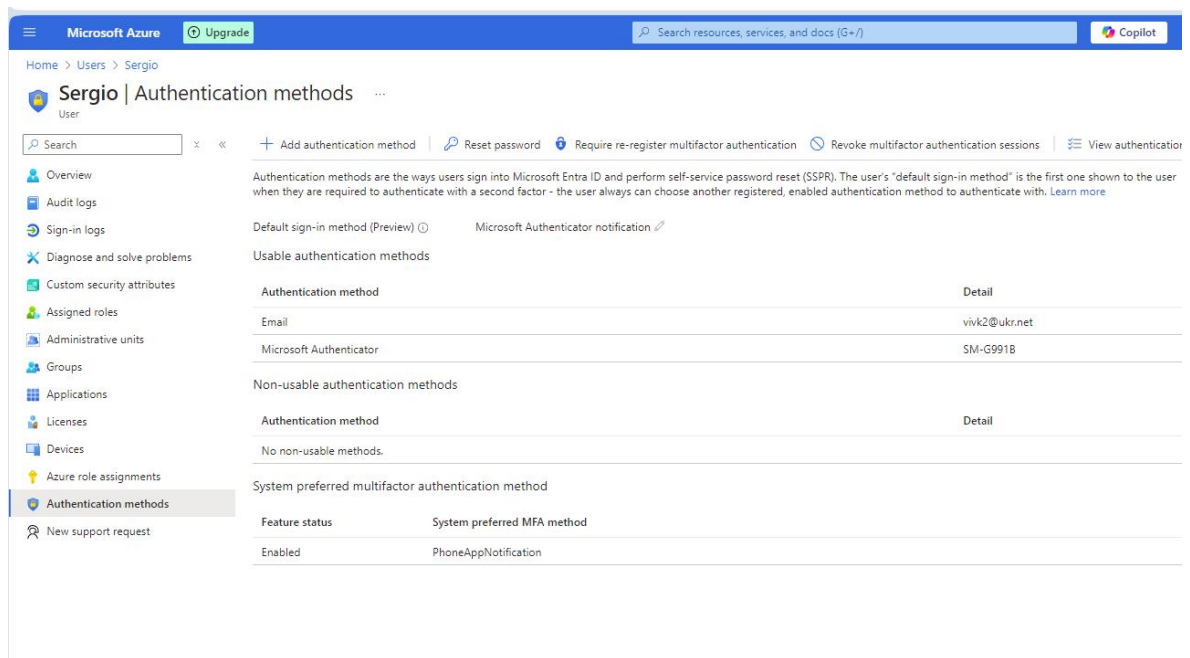
1. Enable Single Sign-On (SSO) for your Microsoft Entra ID tenant.
2. Enforce Multi-Factor Authentication (MFA) for all users in the directory.
3. Configure conditional access policies to require MFA for high-risk sign-ins.
4. Verify that SSO and MFA settings are correctly applied for the users.

Actions Taken:

1. **Single Sign-On (SSO)** by **Password-based** was successfully enabled for Microsoft Entra ID.



2. **Multi-Factor Authentication (MFA)** was activated for all users in the directory.



Configure conditional access policies to require MFA for high-risk sign-ins.

Due to registration limitations for a Premium P2 license (mobile phone number issue), the configuration was performed manually without Premium P2

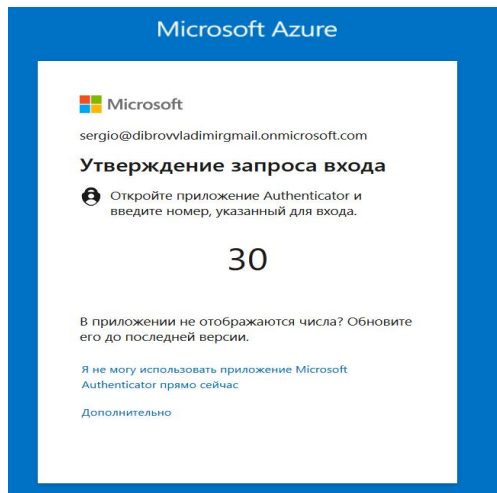
- Configured **MFA status as Enabled** for all users in the directory.
- Navigated to **Azure Active Directory > Security > Conditional Access > Named Locations** to define trusted IP addresses (currently unavailable).
- Ensured MFA is configured for each application individually to enhance security for high-risk sign-ins.
- Used **Subscriptions > Access Control (IAM) > Add Role Assignment** to assign roles requiring MFA for the necessary resources.

Results:

- Achieved a manual setup approximating **high-risk sign-in** scenarios through customized configurations.
- Secured access with **SSO and MFA** functioning effectively.

3. Verification completed:

1. SSO and MFA are functioning as expected, enhancing access security.



Practical Task 3: Implementing Role-Based Access Control (RBAC)

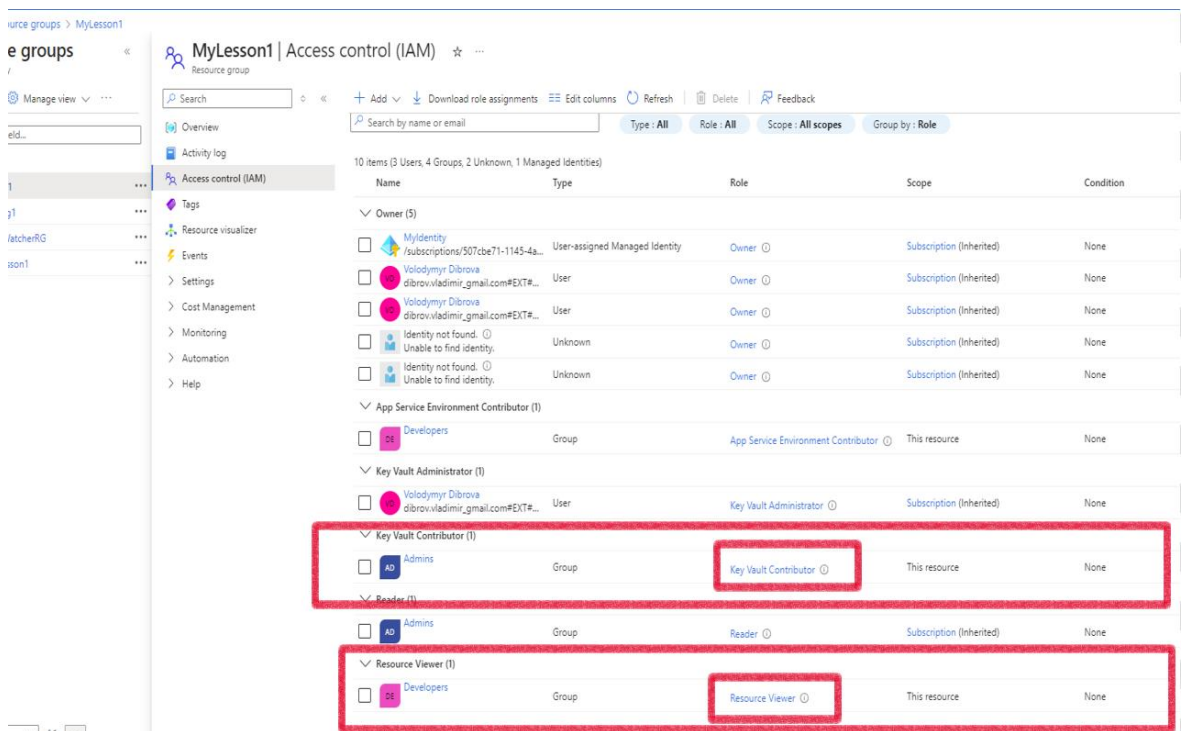
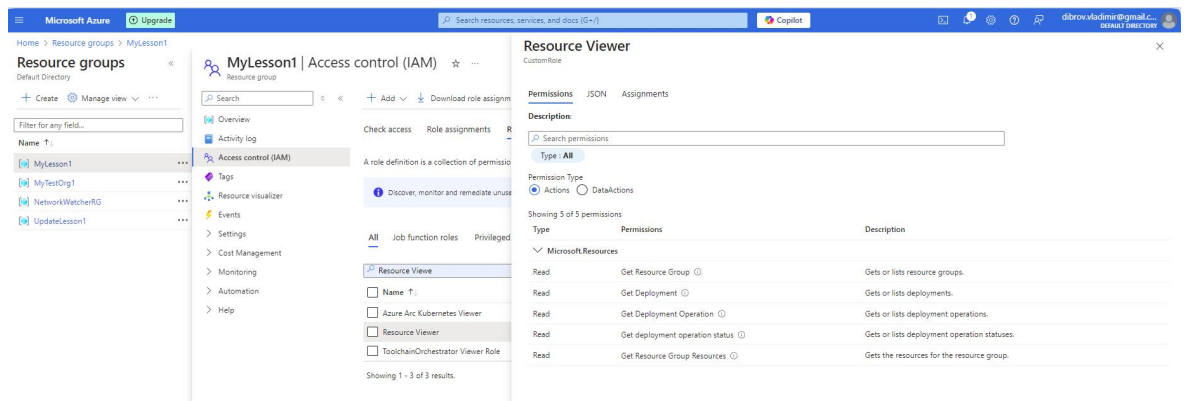
Implement Role-Based Access Control (RBAC) in Azure to manage access to resources based on roles and ensure fine-grained access management.

Requirements:

1. Create a custom role named **Resource Viewer** with read-only permissions for a specific resource group.
2. Assign the **Resource Viewer** role to the **Developers** group created earlier.
3. Assign the built-in **Contributor** role to the **Admins** group for the same resource group.
4. Verify that members of the **Developers** group have only read access and members of the **Admins** group have full access to the resource group.

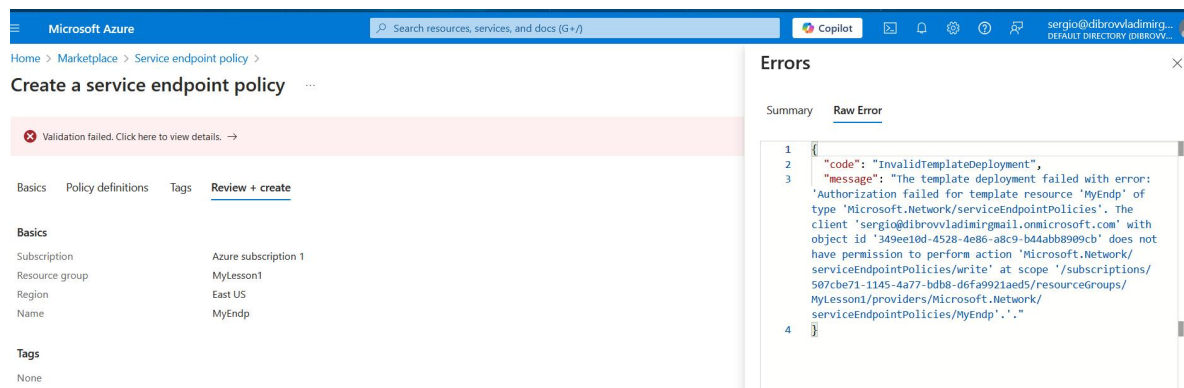
Actions Taken:

1. Created a **custom role Resource Viewer** with read-only permissions for a specific resource group.
2. Assigned the **Resource Viewer** role to the **Developers** group created earlier.
3. Assigned the built-in **Contributor** role to the **Admins** group for the same resource group.



4. Verified access:

1. Members of the **Developers(Sergio)** group have read-only access.
2. Members of the **Admins** group have full access to the resource group.



Results:

- Successfully configured **Role-Based Access Control (RBAC)** to manage resource access based on predefined roles.

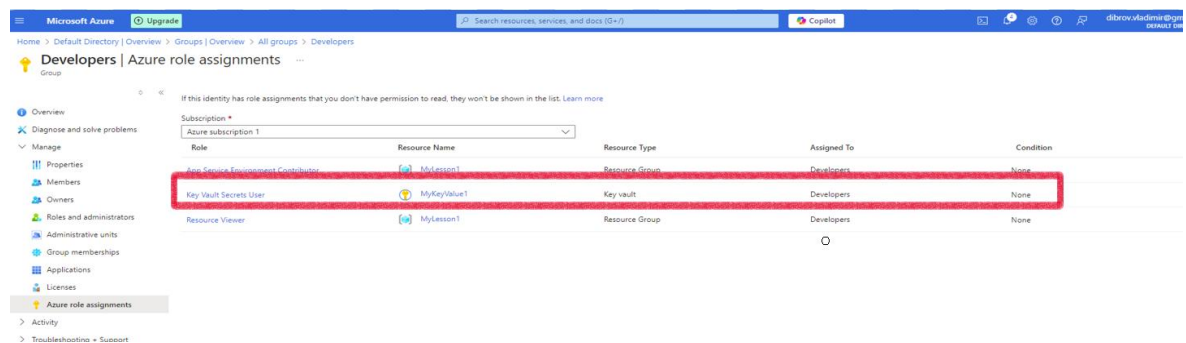
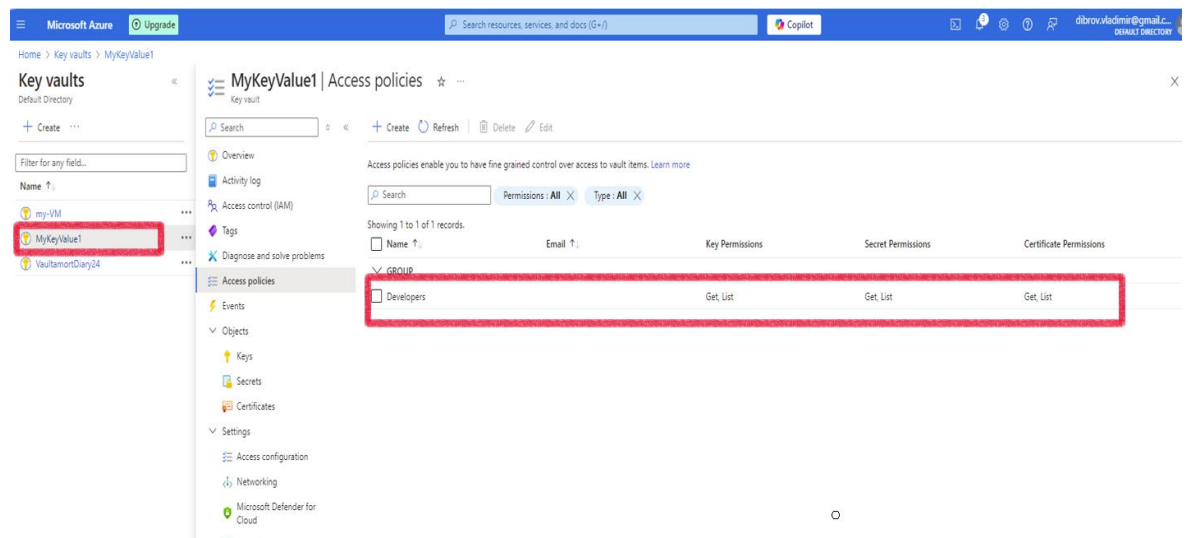
Practical Task 4: Securing Sensitive Information with Azure Key Vault
Set up Azure Key Vault to securely store and manage sensitive information such as keys, secrets, and certificates.

Requirements:

1. Create a new Azure Key Vault in your subscription.
2. Add a secret to the Key Vault (e.g., a database connection string).
3. Set access policies to grant the **Application Developer** role (assigned to the **Developers** group) permission to retrieve secrets from the Key Vault.
4. Verify that only members of the **Developers** group can access the stored secret.

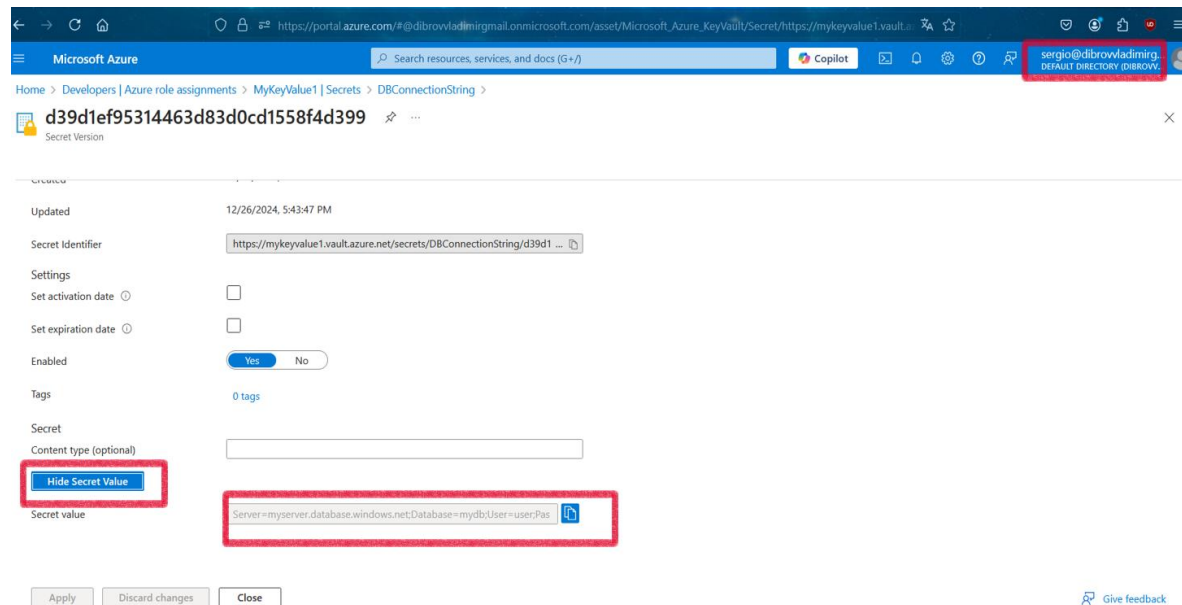
Actions Taken:

1. Created a new **Azure Key Vault** within the subscription.
2. Added a secret to the Key Vault (database connection string).
3. Configured access policies to grant the **Application Developer** role (assigned to the **Developers** group) permission to retrieve secrets from the Key Vault.



4. Verified access:

1. Only members of the **Developers** group can access the stored secret.



Results:

- Successfully set up **Azure Key Vault** to securely store and manage sensitive information, ensuring restricted access based on roles.

Practical Task 5: Creating and Assigning Basic Azure Policies

Define and assign Azure Policies to enforce compliance with organizational standards for resource management.

Requirements:

1. Create an Azure Policy to enforce tagging for all newly created resources with a specific tag.
2. Assign the policy to a resource group.
3. Verify that any new resource created in the resource group without the required tag is marked as non-compliant.
4. Review and document the compliance status of the resource group

Actions Taken:

1. Created an **Azure Policy** to enforce tagging for all newly created resources with a specific tag **Environment: Development**.
2. Assigned the policy to a resource group.

Home > Policy | Assignments > TaggingEnvironment >

TaggingEnvironment

Edit Policy definition

Name * ⓘ

TaggingEnvironment

Description

Category ⓘ

☐ Create new ☒ Use existing

Tags

POLICY RULE

↓ Import sample policy definition from GitHub

🔗 Learn more about policy definition structure

```
1 {
2   "mode": "All",
3   "policyRule": {
4     "if": {
5       "field": "[concat('tags[' , parameters('tagName'), ''])]",
6       "exists": "false"
7     },
8     "then": {
9       "effect": "deny"
10    }
11  },
12  "parameters": {
13    "tagName": {
14      "type": "String",
15      "metadata": {
16        "displayName": "Tag Name",
17        "description": "The name of the tag to check for."
18      }
19    }
20  }
```

3. Verified to create resource without tag “Environment”

Microsoft Azure Upgrade

Search resources, services, and docs (G+)

Copilot

dibrou.vladimir@gmail.com

Home > Resource groups > MyLesson1 > Marketplace > Service endpoint policy >

Create a service endpoint policy

Validation failed. Click here to view details. →

Basics

Policy definitions

Tags

Review + create

Basics

Subscription

Resource group

Region

Name

Azure subscription 1

MyLesson1

East US

MyEndpoint

Tags

None

For this policy to take effect, you will need to associate it to one or more subnets that have virtual network service endpoints. Please visit a virtual network in East US region and then select the subnets to which you would like to associate this policy.

Errors

Summary

Raw Error

ERROR DETAILS

Resource 'MyEndpoint' was disallowed by policy. (Code: RequestDisallowedByPolicy)

Policy: TaggingEnvironment

WAS THIS HELPFUL? 🗣️ 📧

Explain with Copilot

Troubleshooting Options

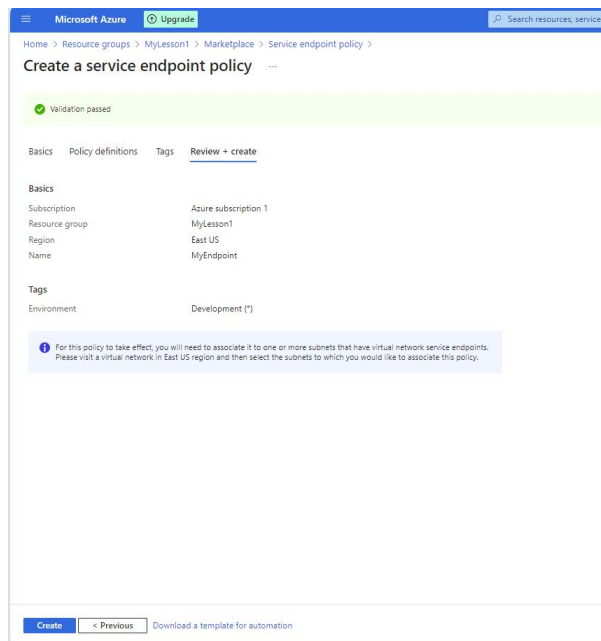
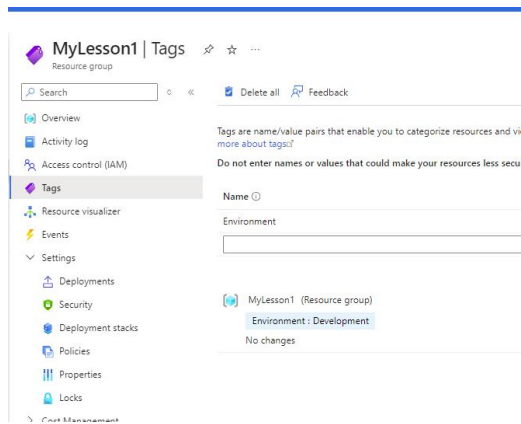
Check Usage + Quota 📄

New Support Request 📄

Give feedback

Tell us about your experience with the ARM Errors page.

4. Verified to create resource with tag “Environment”



Results:

- Successfully implemented an **Azure Policy** to ensure compliance with organizational standards for resource management.

Practical Task 6: Using Policy Effects to Enforce Compliance

Configure Azure Policies with different policy effects to enforce compliance and manage resources according to organizational standards.

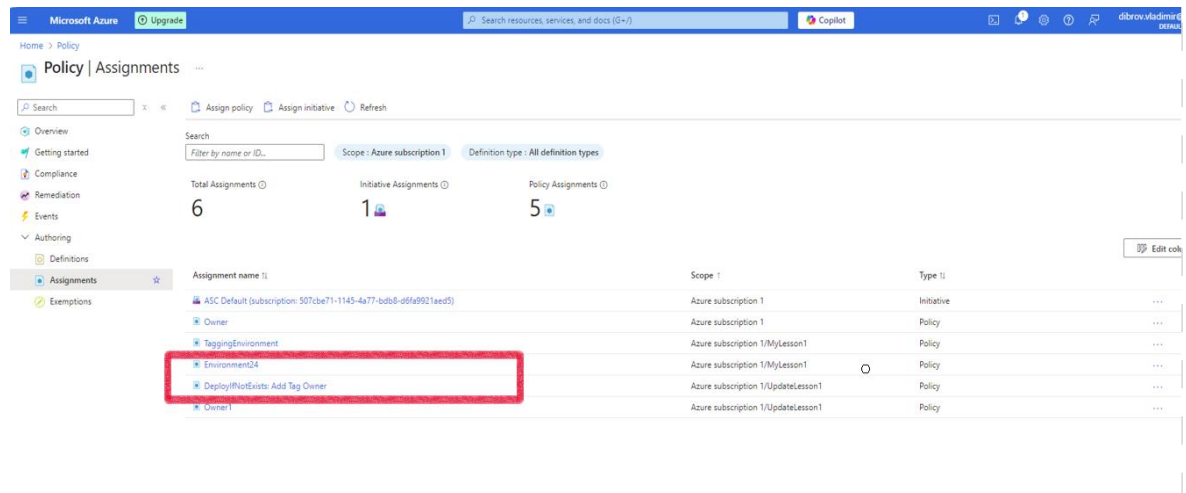
Requirements:

1. Create a policy with the **Audit** effect to monitor and log untagged resources within a resource group.
2. Create a policy with the **DeployIfNotExists** effect to automatically add a specific tag (Owner: IT) to any newly created resource.
3. Assign these policies to a resource group and verify their behavior by:

- o Creating a resource without a tag and checking the compliance logs.
- o Creating a resource to validate the automatic tag deployment.

Actions Taken:

1. Created a policy with the **Audit** effect to monitor and log untagged resources within a resource group.
2. Created a policy with the **DeployIfNotExists** effect to automatically add a specific tag (**Owner: IT**) to any newly created resource.



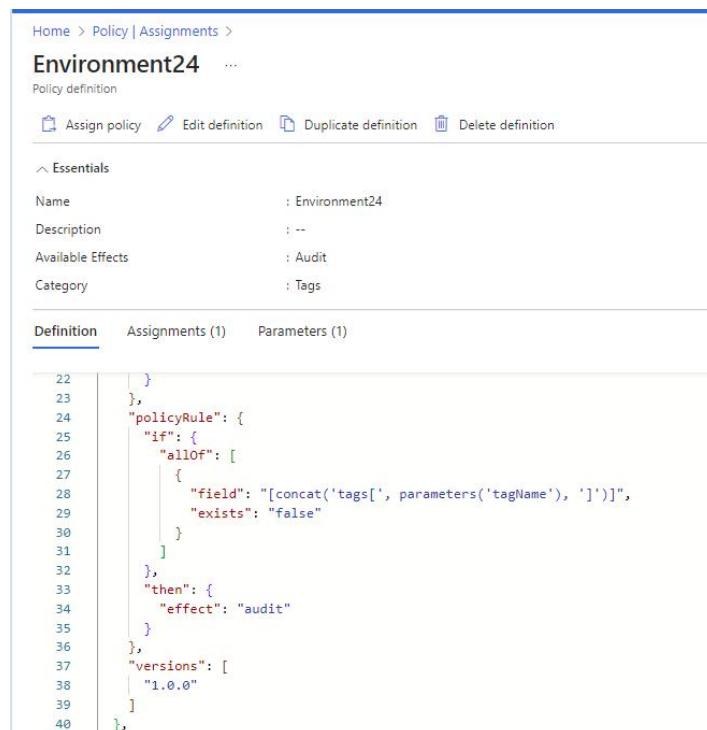
Microsoft Azure | Upgrade | Search resources, services, and docs (Ctrl+K) | Copilot | dltrov.vladimir@... | Home > Policy

Policy | Assignments

Search: Filter by name or ID... Scope: Azure subscription 1 Definition type: All definition types

Total Assignments: 6 Initiative Assignments: 1 Policy Assignments: 5

Assignment name	Scope	Type
ASC Default (subscription: 507cbe71-1145-4a77-bdb8-d6fa9921eed3)	Azure subscription 1	Initiative
Owner	Azure subscription 1	Policy
TaggingEnvironment	Azure subscription 1/MyLesson1	Policy
Environment24	Azure subscription 1/MyLesson1	Policy
DeployIfNotExists: Add Tag Owner	Azure subscription 1/UpdateLesson1	Policy
Owner	Azure subscription 1/UpdateLesson1	Policy



Home > Policy | Assignments > Environment24

Policy definition

Assign policy Edit definition Duplicate definition Delete definition

Essentials

Name: Environment24

Description: --

Available Effects: Audit

Category: Tags

Definition Assignments (1) Parameters (1)

```

22  }
23  },
24  "policyRule": {
25    "if": {
26      "allOf": [
27        {
28          "field": "[concat('tags['', parameters('tagName'), '']')]",
29          "exists": "false"
30        }
31      ]
32    },
33    "then": {
34      "effect": "audit"
35    }
36  },
37  "versions": [
38    "1.0.0"
39  ]
40  },

```

Home > Policy | Assignments >

DeployIfNotExists: Add Tag Owner

Policy definition

Assign policy Edit definition Duplicate definition Delete definition

Essentials

Name: DeployIfNotExists: Add Tag Owner

Description: This policy automatically adds the Owner: IT tag to resources without the tag.

Available Effects: Modify

Category: Tags

Definition Assignments (1) Parameters (1)

```

22     },
23   },
24   "policyRule": {
25     "if": {
26       "allOf": [
27         {
28           "field": "[concat('tags['', parameters('tagName'), '']')]",
29           "exists": "false"
30         }
31       ]
32     },
33     "then": {
34       "effect": "modify",
35       "details": {
36         "roleDefinitionIds": [
37           "/providers/Microsoft.Authorization/roleDefinitions/8e3af657-a8ff-443c-a75c-2fe8c4cb635"
38         ],
39         "operations": [
40           {
41             "operation": "addOrReplace",
42             "field": "[concat('tags['', parameters('tagName'), '']')]",
43             "value": "IT"
44           }
45         ]
46       }
47     }
48   },
49   "success": true

```

Microsoft Azure Upgrade Search resources, services, and docs (Ctrl) Copilot

Home > Policy

Policy | Compliance

Search Assign policy Assign initiative Refresh

Overview Getting started Compliance Remediation Events Authoring Definitions Assignments Exemptions

Search Filter by name or ID... Scope: Azure subscription 1 Definition type: All definition types Compliance state: All compliance states

Overall resource compliance 13% 1 out of 8

Resources by compliance state 8

Compliant 7 Non-compliant 1

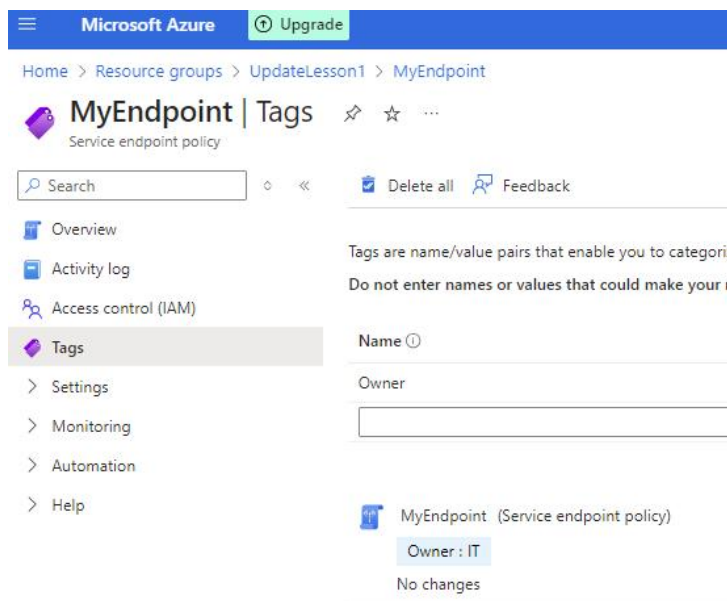
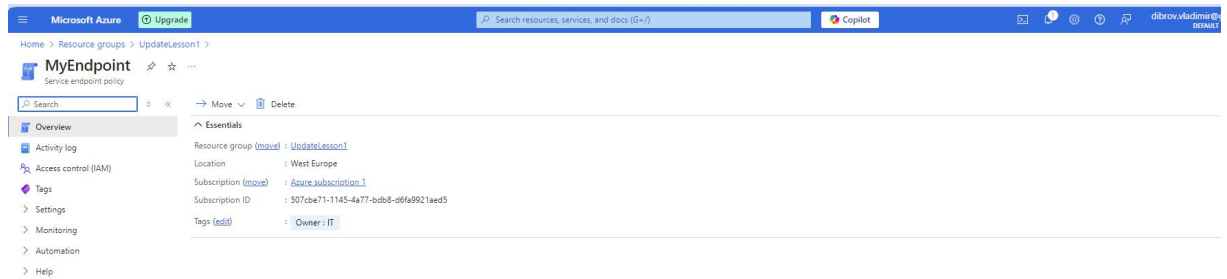
Non-compliant initiatives 1 out of 1

Non-compliant policies 25 out of 233

Name	Scope	Compliance state	Resource compliance	Non-compliant resources	Non-compliant policies
ASC Default (subscription: 307cbe71-1143-4a77-bdb8-d8)	Azure subscription 1	Non-compliant	0% (0 out of 5)	5	23
TaggingEnvironment	Azure subscription 1/MyLesson1	Non-compliant	0% (0 out of 4)	4	1
Environment24	Azure subscription 1/MyLesson1	Non-compliant	0% (0 out of 4)	4	1
DeployIfNotExists: Add Tag Owner	Azure subscription 1/UpdateLesson1	Compliant	100% (1 out of 1)	0	0
Owner1	Azure subscription 1/UpdateLesson1	Compliant	100% (0 out of 0)	0	0
Owner	Azure subscription 1	Compliant	100% (0 out of 0)	0	0

3. Assigned both policies to a resource group and verified their behavior:

1. Created a resource without a tag and checked the compliance logs for auditing.
2. Created a resource to confirm the automatic deployment of the specified tag.



Results:

- Successfully configured and tested **Azure Policies** with different effects.